

UNIVERSIDAD AUTONOMA DE MADRID

ESCUELA POLITECNICA SUPERIOR



Grado en Ingeniería Informática

TRABAJO FIN DE GRADO

**Herramienta de extracción y visualización de grandes datos de
cursos MOOC**

Autor: Alberto Velasco López de Ochoa

Tutor: Germán Montoro Manrique

MAYO 2018

Herramienta de extracción y visualización de grandes datos de cursos MOOC

AUTOR: Alberto Velasco López de Ochoa

TUTOR: Germán Montoro Manrique

Grupo de Herramientas Interactivas Avanzadas

Dpto. Ingeniería Informática

Escuela Politécnica Superior

Universidad Autónoma de Madrid

Mayo de 2018

Resumen (castellano)

La interpretación y toma de decisiones sobre los datos recolectados de un sistema informático es, en muchas ocasiones, una tarea poco intuitiva y no trivial. La gran cantidad de datos que se deben analizar hace difícil, a simple vista, llegar a obtener información realmente útil de tu plataforma.

Este problema afecta a todos los sectores que decidan almacenar sus datos para su futuro estudio, desde el comercial, al deportivo, incluso al educacional. Debido a esto, es necesaria una herramienta que procese dichos datos y los muestre de una forma mucho más útil para el usuario.

Lo cursos online masivos abiertos, del inglés Massive Open On-line Course (MOOC) no son una excepción; y la UAM se encontraba en la misma situación con sus cursos hospedados en edX.

Abordando estos problemas, en este Trabajo de Fin de Grado se plantea desarrollar una herramienta web auxiliar para poder extraer, visualizar los datos de dichos cursos y poder realizar consultas fijadas para obtener información útil de la plataforma.

Palabras clave (castellano)

Sistema informático, MOOC, edX, herramienta web, información útil.

Abstract (English)

The interpretation and decision-making of the data collected on a computer system is, usually, counter-intuitive but of a huge importance. The large amount of data which must be analyzed makes it difficult to obtain relevant information about your platform at a simple glance.

This problem affects every single sector that requires the collection of data for its future studying, from the commercial point of view to the educational or even the sportive. Given this situation, it is necessary to develop a tool that can process this data and show it in a way that can be more useful for the user. The Massive Open On-line Courses (MOOC) are not an exception; and the UAM founded itself on the exact same situation regarding its courses hosted on edX.

By tackling these problems, this Final Degree Project brings up the development of a web tool that can help to draw out, visualize the data of the aforementioned courses and, by doing so, being able to conduct pre-established enquiries in order to obtain useful information regarding the platform.

Keywords (English)

Computer system, MOOC, edX, web tool, useful information.

Agradecimientos

La lista de gente a la que tengo algo que agradecer es enorme ya que, hoy día, estoy donde estoy gracias a ellas. Gracias a mis compañeros en la EPS. Con ellos he aprendido, reído, peleado y afrontado todos estos años de estudio, y sin ellos no habría sido la misma experiencia. Agradezco también a mi tutor, Germán, todo el trabajo que ha hecho por mí en este largo proyecto, esos tropocientos emails que te mandaba preguntando y poniéndote al día y tú respondiéndome fuese el día que fuese.

Gracias también a mi pareja, Alejandra, por apoyarme y estar conmigo en los buenos y malos momentos. Por escuchar mis frustraciones y alegrías y saber aconsejarme cuando veía las cosas de forma pesimista.

A Antonio, Juan y Juanjo les dedico mi más profundo “gracias”. Vuestra amistad me ha llevado a estar donde estoy y ser quien soy.

Y, sobre todo, quiero agradecerse a mis padres. Por tener esa paciencia infinita conmigo a lo largo de todos estos años. A mi padre, que me enseñó el valor de la constancia y trabajo duro, aquellas charlas post-examen analizando juntos los resultados e intentando ver que nota podía sacar. A mi madre, porque de ella aprendí que las oportunidades hay que aprovecharlas y hay que sacar el pundonor interno de donde sea en ciertas ocasiones. Nunca podré agradecerlos lo suficiente lo mucho que habéis hecho por mí, esperando nada a cambio.

A todos vosotros, gracias.

INDICE DE CONTENIDOS

1 Introducción	1
1.1 Motivación.....	1
1.2 Objetivos.....	1
1.3 Organización de la memoria.....	1
2 Estado del arte	2
2.1 MOOC	2
2.2 Plataforma edX	3
2.3 Learning Analytics	3
3 Diseño.....	4
3.1 Estado del sistema antes del proyecto	4
3.2 Roles de Usuario.....	5
3.3 Pre-Requisitos.....	5
3.4 Requisitos Funcionales	6
3.4.1 Gestión de los archivos.....	6
3.4.2 Gestión de los cursos	7
3.4.3 Gestión de consultas	7
3.5 Requisitos No Funcionales	7
3.6 Casos de Uso	8
3.7 Diagrama de la BBDD.....	9
4 Desarrollo	10
4.1 Análisis de Datos.....	10
4.2 Prototipo inicial	11
4.3 Aplicación final	14
5 Integración, pruebas y resultados	17
5.1 Integración.....	17
5.2 Pruebas y resultados	21
6 Conclusiones y trabajo futuro	24
6.1 Conclusiones.....	24
6.2 Trabajo futuro	25
Referencias	27
Glosario	29
Anexos.....	I
A Formato de los ficheros Certificate y Profile.....	I
B Estructura de las tablas auxiliares de la BBDD.....	II
C Ficheros del componente web	III

INDICE DE FIGURAS

FIGURA 1: TIPOS DE MOOC'S Y PLATAFORMAS SEGÚN SU OBJETIVO [4]	2
FIGURA 2: ESTRUCTURA DEL DIRECTORIO DE TRABAJO	4
FIGURA 3: DIAGRAMA DE CASOS DE USO DE LA APLICACIÓN	8
FIGURA 4: ESQUEMA DE LA BBDD	9
FIGURA 5: PROCESO DE CARGA DE LOS CURSOS.	12
FIGURA 6: ESQUEMA DE LA ESTRUCTURA DE LA WEB EN EL PROTOTIPO	13
FIGURA 7: FLUJO DE LA NUEVA VERSIÓN DE CARGAR CURSO	14
FIGURA 8: MATRIZ DE DECISIÓN.....	15
FIGURA 9: PÁGINA DE INICIO DE LA APLICACIÓN.....	17
FIGURA 10: PRIMERA PARTE DEL PROCESO DE BORRAR LOS CURSOS.....	17
FIGURA 11: MENSAJE INFORMATIVO AL TERMINAR EL PROCESO DE BORRAR LOS CURSOS.	18
FIGURA 12: PANTALLA PARA CARGAR UN CURSO.....	18
FIGURA 13: MENSAJE INFORMATIVO DE LA CARGA DE CURSOS.	19
FIGURA 14: MENÚ DE CONSULTAS	19
FIGURA 15: FRAGMENTO DE LA TABLA GLOBAL DE TODOS LOS CURSOS.	20
FIGURA 16: FRAGMENTO DE LAS CONSULTAS DEMOGRÁFICAS.....	20
FIGURA 17: EJEMPLO DE CONSULTA DEMOGRÁFICA.....	21
FIGURA 18: EJEMPLO DE CONSULTA TIPO RANKING	21
FIGURA 19: SCRIPT DE PRUEBA DE ESTRÉS.	22
FIGURA 20: RESULTADOS DE LA CARGA DE 100 CURSOS.....	22
FIGURA 21: ESTUDIO SOBRE EL NIVEL DE ESTUDIOS GENERAL DE TODOS LOS CURSOS.	23
FIGURA 22: ESTUDIO SOBRE EL RANKING DE LOS CURSOS MÁS ESTUDIADOS.	23
FIGURA 23: QUERY PLAN DE LA 1ª CONSULTA.....	23
FIGURA 24: QUERY PLAN DE LA 2ª CONSULTA.....	23

FIGURA 25 DE ANEXO A: FICHERO PROFILE.CSV DEL CURSO ANDROID.	I
FIGURA 26 DE ANEXO A: FICHERO CERTIFICATE.CSV DEL CURSO ANDROID.....	I
FIGURA 27 DE ANEXO B: TABLA AUXILIAR GÉNERO.....	II
FIGURA 28 DE ANEXO B: TABLA AUXILIAR NIVEL ESTUDIOS	II
FIGURA 29 DE ANEXO B: TABLA AUXILIAR DE PAÍSES.....	II
FIGURA 30 DE ANEXO C: DIRECTORIO DE LA APLICACIÓN UAMX DEL SERVIDOR XAMPP.	III

1 Introducción

1.1 Motivación

La interpretación y toma de decisiones sobre los datos recolectados de un **sistema informático** es, en muchas ocasiones, una tarea poco intuitiva y no trivial. La gran cantidad de datos que se deben analizar hace difícil, a simple vista, llegar a obtener información realmente útil de tu plataforma.

Este problema afecta a todos los sectores que decidan almacenar sus datos para su futuro estudio, desde el comercial, al deportivo, incluso al educacional. Debido a esto, es necesaria una herramienta que procese dichos datos y los muestre de una forma mucho más útil para el usuario.

En esta misma situación se encontraba la UAM (a partir de ahora, el cliente) con la gestión de sus cursos **MOOC**. Disponían de una gran cantidad de datos sobre sus usuarios y sus cursos, pero no disponían de una herramienta para poder extraer esos datos, procesarlos y analizarlos. Fue entonces necesario, acordar una reunión con el cliente para estipular **qué** querían y **cómo** lo querían. Se necesitaba una aplicación donde poder consultar todos los datos de todos los cursos en una única tabla, normalizando y reparametrizando ciertos datos y poder hacer consultas sobre dichos cursos. Siempre siguiendo el marco legal de la LOPD (Ley Orgánica de Protección de Datos), anonimizando los datos de carácter sensible de los usuarios.

La principal motivación personal para desarrollar este proyecto fue precisamente el análisis de datos. Debido a la importancia y repercusión que tiene actualmente este sector en la industria informática, este proyecto podría servir como una primera prueba de contacto con este tipo de aplicaciones, aunque sea de una forma mucho menos ambiciosa.

1.2 Objetivos

El objetivo del presente proyecto es el desarrollo de una aplicación o herramienta de soporte para la extracción de datos de los cursos MOOC que tiene la UAM actualmente, procesarlos y visualizarlos a través de la aplicación con un formato definido por el cliente y poder realizar una serie de consultas fijas sobre esos datos.

1.3 Organización de la memoria

El resto del documento se estructura como se indica a continuación:

- En el apartado 2 se describe de forma general los MOOC y la plataforma edX, así como describir un poco la importancia de Learning Analytics en este tipo de plataformas.
- En el apartado 3 se explica el Diseño de la aplicación, así como la toma de decisiones sobre los diferentes tipos de tecnologías usadas en el proyecto
- En el apartado 4 se detalla la implementación del sistema.
- En el apartado 5 se expondrán los resultados obtenidos tras la integración y realización de pruebas sobre la aplicación.
- Por último, en el apartado 6 se expondrán las conclusiones y trabajo futuro del proyecto.

2 Estado del arte

En esta sección se presenta una revisión del estado del arte sobre los MOOC's, como también sobre una de las principales plataformas de cursos masivos online que da soporte a los cursos que oferta la UAM y para finalizar se comentará la necesidad y ventajas de usar Learning Analytics sobre la información recogida en un sistema de gestión de cursos, con sus consecuentes datos.

2.1 MOOC

Los **MOOC's** (Massive Open On-line Course) también conocidos como **COMA** en español, son cursos abiertos en línea enfocados a una grandísima cantidad de participantes accesibles a través de Internet [1]. Un MOOC, en el sentido más estricto de la palabra, es un curso con sus lecciones, su contenido multimedia, exámenes; pero también es mucho más. Es un evento en el cual, gente interesada en un mismo tema decide reunirse, conectar y colaborar unos con otros para así desarrollar una mejor actividad educacional.

El término MOOC apareció por primera vez en 2008 de la mano de los activistas educacionales **Dave Cormier** y **Bryan Alexander** los cuales iniciaron un curso online llamado "Connectivism and Connective Knowledge" [2] el cual tuvo 2300 participantes interesados. Anteriormente, en el año 2000 ya había habido intentos de realizar este tipo de cursos, pero no proliferaron debido a que Internet todavía no estaba tan extendido.

En la actualidad, se pueden definir dos tipos, o enfoques, de MOOC's [3]:

- **xMOOC**: se centra en la figura del "experto", el profesor se convierte en la figura principal del aprendizaje, siendo él quien crea y coordina el plan de estudios. Los alumnos reciben el material facilitado por la plataforma y tiene la responsabilidad de realizar las tareas asignadas en las fechas asignadas. Tiene una duración definida y al finalizar se emite un certificado.
- **cMOOC**: el enfoque de este tipo de cursos se centra en la adquisición y generación del conocimiento por parte de los participantes. No hay un plan de estudios definido, y se ofrece una serie de herramientas a través de la web (wikis, blogs, plataforma de redes sociales, ...) a través de los cuales los alumnos contribuirán al aprendizaje colectivo y se irán autoevaluando. La figura central, es, por tanto, el alumno.

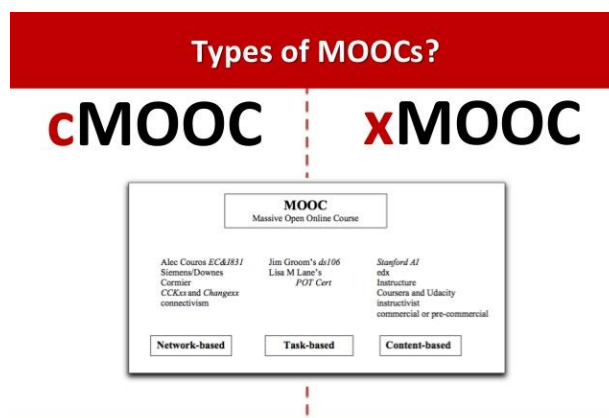


Figura 1: Tipos de MOOC's y plataformas según su objetivo [4]

2.2 Plataforma edX

En la actualidad, existen multitud de plataformas de cursos online masivos; entre ellas edX. Vamos a hablar de esta plataforma en concreto, debido a que la UAM colabora con esta para hospedar los cursos que oferta la universidad.

edX¹ fue fundada en 2012 por la Universidad de Harvard y el Instituto Tecnológico de Massachusetts con el objetivo de proveer MOOC's universitarios de un amplio abanico de disciplinas a estudiantes de todo el mundo. Hoy en día edX es una de las plataformas más importantes contando con 4 millones de usuarios y siendo la segunda plataforma más importante a nivel mundial [5].

Esta plataforma usa un enfoque xMOOC para sus cursos, facilitando el material necesario al alumno, por parte del profesorado, para avanzar en su aprendizaje. La UAM, que en la actualidad tiene 14 cursos en dicha plataforma, ofrece un amplio catálogo de cursos, abarcando multitud de áreas: programación Android, biología, humanidades, ... Cada curso puede tener varias ediciones, dependiendo de cuantas veces se repita el mismo.

Pero edX no solo ofrece una plataforma para hospedar cursos, sino que también facilita a las diferentes universidades los archivos ² de las actividades que se registran dentro de sus cursos y los datos personales de los usuarios registrados en tus cursos. Dicha información puede suponer un recurso muy valioso para las universidades, pudiendo saber qué mejorar o que mantener de su método educativo.

2.3 Learning Analytics

Debido a que disponemos de los datos recolectados por edX sobre los usuarios y los eventos de nuestros cursos, se debería proceder a analizarla para así obtener información útil sobre nuestra plataforma. La disciplina que se dedica a analizar los datos de sistemas informáticos basados en la educación se llama **Learning Analytics**, o análisis del aprendizaje en español.

El software utilizado normalmente para el análisis de estos datos es propio del entorno virtual de aprendizaje, aunque se suelen usar herramientas auxiliares. Aún que no se haya usado ninguna de estas herramientas en el trabajo, debido al carácter de este, es conveniente conocer al menos las más importantes:

- SNAPP³ es una herramienta que permite ver la red de interacciones que se origina a través de los mensajes en los foros de discusión y de ejercicios. Sirve para identificar patrones de comportamiento. Este tipo de software se usa, por ejemplo, en Moodle.
- LOCO-ANALYST⁴ permite analizar los procesos que suceden en ambiente de aprendizaje. Esta herramienta se usa para entornos web. Esta herramienta hace más hincapié en qué actividades y ejercicios hacen los estudiantes dentro de la plataforma.

¹ <https://www.edx.org>

² La documentación para descargar y saber la estructura para el posterior análisis se puede consultar en <http://edx.readthedocs.io/projects/devdata/en/latest/>

³ <http://ww38.snappvis.org/>

⁴ <http://jelenajovanovic.net/LOCO-Analyst/>

3 Diseño

3.1 Estado del sistema antes del proyecto

Todo el trabajo, tanto previo como durante el proyecto, se ha realizado en uno de los ordenadores facilitados por la Unidad de Tecnologías para la Educación (UTED) de la UAM. Para contextualizar el proyecto y explicar ciertas decisiones de diseño, procederemos a explicar la estructura tanto de hardware y software del sistema:

- **Hardware:** ya hemos mencionado previamente, es un ordenador localizado en las oficinas de la UTED de la UAM. Lo reseñable es que todo el trabajo se ha realizado en dicho ordenador debido a la confidencialidad de los datos de los usuarios de la plataforma.
- **Software:** el sistema operativo de 64 bits del ordenador es *Windows 7 Professional*. Dispone de bastantes scripts personales, usados para labores propias del grupo de cursos MOOC, siendo Python el lenguaje de programación. PostgreSQL es el gestor de BBDD principal. Como servidor web se usa XAMPP. Como dato adicional, se ha usado TeamViewer para posibilitar el trabajo telemático.

El sistema actual cuenta con una serie de scripts encargados de la gestión, descarga de documentos y obtención de la información de los cursos. El directorio donde se centrará el trabajo previo al desarrollo del proyecto (y donde se encuentran los scripts anteriormente mencionados) es D:\Czar del sistema. En dicho directorio, hay una serie de carpetas, cada una almacena datos de los cursos. A continuación, mostramos un esquema indicando el uso concreto de cada directorio:

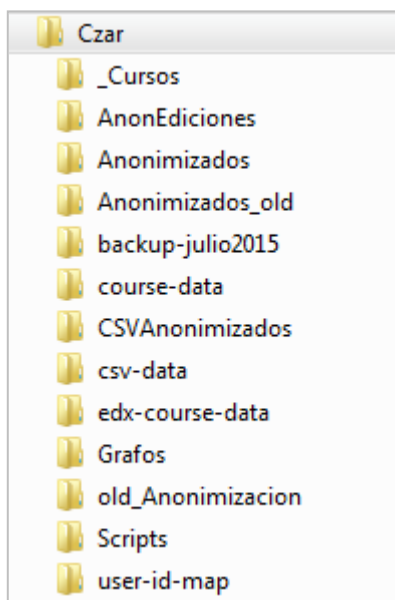


Figura 2: Estructura del directorio de trabajo

No hablaremos de todos los directorios mostrados en la Figura 2, ya que algunos están obsoletos, sino que se explicarán aquellos que tienen relevancia para el proyecto. Los más importantes serían:

- **AnonEdiciones:** dispone de los ficheros events anonimizados de los cursos. Es necesaria esta anonimización para seguir el marco legal de la Ley de Protección de Datos.
- **Course-data:** contiene todos los ficheros, sin tratar, descargados de edX.
- **Csv-data:** directorio clave. Contiene los .csv que se usaran en el proyecto, ya anonimizados, para obtener los datos que desea el cliente. Por cada curso se generan tres ficheros .csv. De estos, solo necesitamos Profile y Certificate

(Profile contiene los datos demográficos de los estudiantes del curso y Certificate la nota e información para expedir la titulación).

- **Edx-course-data:** contiene todos los logs por días de los usuarios en los cursos de la plataforma.
- **Scripts:** contiene todos los ejecutables .py necesarios para obtener todos los archivos mencionados anteriormente. Es el núcleo de toda la estructura de trabajo.

Como hemos comentado, el núcleo del trabajo es el directorio Scripts en el que hay una serie de scripts para obtener los ficheros que contienen la información relativa a la **última semana** de actividad de los cursos. Todos los scripts están programados en Python. Dichos ficheros resultantes a la ejecución deben de sufrir un proceso de anonimización, además de procesarlos para obtener la información en el formato adecuado.

Debido a que este proceso de ejecución de los scripts no es el tema principal del proyecto, todo este proceso, viene explicado y recogido en dos pdf⁵ facilitados por el propio cliente. Se detallan la estructura de datos y los scripts a usar, como editarlos y el proceso a seguir para obtener los archivos finales que obtienen los datos de los cursos. Los archivos con la información de los usuarios de la última semana serán el **punto de partida** de este proyecto.

3.2 Roles de Usuario

En la aplicación desarrollada solo existe un tipo de usuario final (miembros del grupo de cursos MOOC de la UAM), que será quien haga consultas sobre los cursos añadidos. Dichos usuarios deben tener conocimiento previo de cómo obtener los archivos previos necesarios para el funcionamiento de la aplicación.

3.3 Pre-Requisitos

Debido a que hay una serie de precondiciones críticas para el correcto desarrollo y funcionamiento del proyecto, a continuación, se detallarán todas las condiciones previas que se deben de dar:

PRE1: el dispositivo sobre el que se desarrollará la aplicación y ejecutarán las pruebas de rendimiento será el ordenador facilitado por el grupo de cursos MOOC de la UAM.

PRE2: para el correcto funcionamiento de la aplicación desarrollada en este proyecto, se deberán de contar previamente con los archivos .csv especificados en el apartado 3.1 en la sección de csv-data.

PRE2.1: los archivos Profile y Certificate de cada curso deberán estar bien formados. En la primera línea del documento deberán aparecer (separados por “;”) los diferentes nombres campos del fichero. En el Anexo A hay un ejemplo de ficheros bien formados tanto para el archivo Profile como el Certificate facilitados por el cliente.

PRE2.2: para obtener dichos archivos, aunque ya se comentó en la sección 3.1 de esta memoria, se debe proceder a la ejecución de los scripts indicados en los pdf’s facilitados por el cliente.

PRE3: todo archivo que tenga información personal de carácter sensible del usuario debe ser anonimizado.

⁵ Dichos pdf’s se llaman “Guía de Iniciación a los datos de Edx” y “Guía de actualización de datos”. El primero habla sobre la estructura de los datos y el segundo sobre el procedimiento de ejecución. Se encuentra, en la ruta D:\usuarios\UAMx\Desktop\GuíaCzar.

PRE4: el ordenador deberá contar con una serie de programas ya instalados para poder ejecutar la aplicación desarrollada. Deberá contar con Python, PostgreSQL y XAMPP.

PRE4.1: La versión instalada de Python deberá ser 2.7.

PRE4.2: Deberán estar creadas dos bases de datos dentro de PostgreSQL para el correcto funcionamiento de la aplicación: “tfg_uamx” (BBDD principal) y “tfg_auxiliar” (BBDD encargada de funciones auxiliares como borrar).

PRE4.3: La versión de XAMPP a utilizar será la última existente.

3.4 Requisitos Funcionales

A continuación, se detallarán los requisitos funcionales obtenidos de la reunión con el cliente el día 20 de diciembre de 2017. Aunque no hay ningún documento firmado en el cual el cliente da expresamente su visto bueno a los requisitos enumerados a continuación, el tutor del proyecto estuvo en dicha reunión, además que al finalizar se expusieron los requisitos captados y el cliente dio su aprobación.

3.4.1 Gestión de los archivos

[RF1] Archivos necesarios: los archivos finales a utilizar en el sistema, una vez descargados todos los necesarios, para conseguir los datos que se necesitan para la aplicación son: Certificate<NombreDelCurso>.csv y Profile<NombreDelCurso>.csv.

[RF2] Selección de los datos de los archivos: la información de utilidad para el cliente, de los ficheros mencionados en RF1, se encuentran recogidas en las siguientes columnas de cada fichero⁶:

- Profile: user_id, gender, year_of_birth, level_education, goals, allow_certificate, country.
- Certificate: user_id, grade.

[RF3] Formato de los datos: algunos de los datos obtenidos de los archivos tienen que ir formateados⁷ según indicó el cliente:

- Profile
 - Gender: asignar un valor numérico a cada uno de los valores de este campo.
 - Year: este campo tiene el año de nacimiento del estudiante, pero se quiere que se muestre su edad (en años).
 - Level_education: asignar un valor numérico a cada uno de los valores de este campo.
 - Country: asignar un valor numérico a cada uno de los valores de este campo.

⁶ En el Anexo A se muestran todos los campos originales de dichos archivos. En la reunión con el cliente, fue él quien valoró la elección de todos estos campos debido a la importancia que tenían para sus estudios.

⁷ En el Anexo B se pueden ver estos formateos mejor debido a que ahí se muestran las tablas auxiliares ya formateadas con los formatos solicitados.

3.4.2 Gestión de los cursos

[RF4] Cargar curso: el sistema deberá permitir cargar la información descargada previamente de los cursos a la aplicación.

Entrada: El usuario que añada un curso, deberá introducir los siguientes datos:

- Nombre que se desea darle al curso dentro de la aplicación
- Archivo Profile.csv del curso a añadir.
- Archivo Certificate.csv del curso a añadir.

Proceso: el sistema procesará dichos archivos y almacenará los datos en la base de datos, asignándole al curso un id único dentro del sistema.

Salida: si el curso se ha añadido correctamente, aparecerá un mensaje en la interfaz informando que todo ha funcionado correctamente y sino, aparecerá un mensaje de error indicando la incidencia y no añadiéndose el curso a la base de datos.

[RF5] Borrar cursos: el sistema permitirá, una vez realizadas las operaciones estimadas, borrar todos los cursos cargados de la última semana.

Proceso: el sistema borrará todos los cursos cargados de la base de datos, quedando la aplicación “limpia” para una nueva carga de cursos desde cero.

Salida: a través de la interfaz, aparecerá un mensaje informando al usuario del éxito de la operación.

3.4.3 Gestión de consultas

[RF6] Consultar todos los cursos cargados: el sistema deberá permitir, una vez cargados todos los cursos deseados, mostrar en una única tabla con los campos que solicitó el cliente en el RF2 con todos los cursos, identificando cada curso diferente con un id único.

Proceso: mediante una consulta sql contra la base de datos, se obtiene la tabla con todos los campos solicitados y se devuelve a la aplicación.

Salida: se obtiene una tabla con todos los campos mencionados en el RF2 y con un campo extra “id_course” para identificar cada curso cargado.

[RF7] Realizar consultas fijas: el sistema deberá ofrecer una serie de consultas fijadas por el desarrollador en las cuales se podrá obtener información de los diferentes cursos.

Entrada: dependiendo del tipo de consulta, se seleccionará la opción a consultar que el usuario quiera (país, curso, genero, edad, etc.).

Proceso: se realizará una consulta sql contra la base de datos y se devuelve a la aplicación.

Salida: se obtiene el resultado de la consulta seleccionada.

[RF7.1] Consultas demográficas: una parte de las consultas fijas son de carácter demográfico para observar las diferencias en los cursos de forma más clara.

[RF7.2] Consultas en formato “ranking”: la otra parte de las consultas debían estar enfocadas a obtener la información de forma ordenada similar a un ranking (ejemplos: cursos más estudiados, país con más alumnos en la plataforma, etc.).

3.5 Requisitos No Funcionales

Seguidamente, se exponen los requisitos no funcionales de la aplicación; aquellos requisitos que, aunque no esté especificada una funcionalidad concreta del software, sí detallan características del tipo: usabilidad, rendimiento, disponibilidad, ...

[RNF1] Uso de una interfaz gráfica para la aplicación: enfocado a mejorar la usabilidad, se contará con una interfaz gráfica sencilla que facilitará a los usuarios el uso de la aplicación.

[RNF2] Mensajes informativos: antes y después de realizar acciones de importancia (cargar curso, borrar cursos, consultar) la aplicación mostrará mensajes explicativos e informativos para ayudar al usuario o corregirle en caso de error.

[RNF3] Velocidad de carga de los cursos en la aplicación y consultas de estos: la carga cada uno de los cursos en la aplicación no debe superar un lapso de tiempo elevado (más de 20 segundos en caso de cursos con gran cantidad de usuarios).

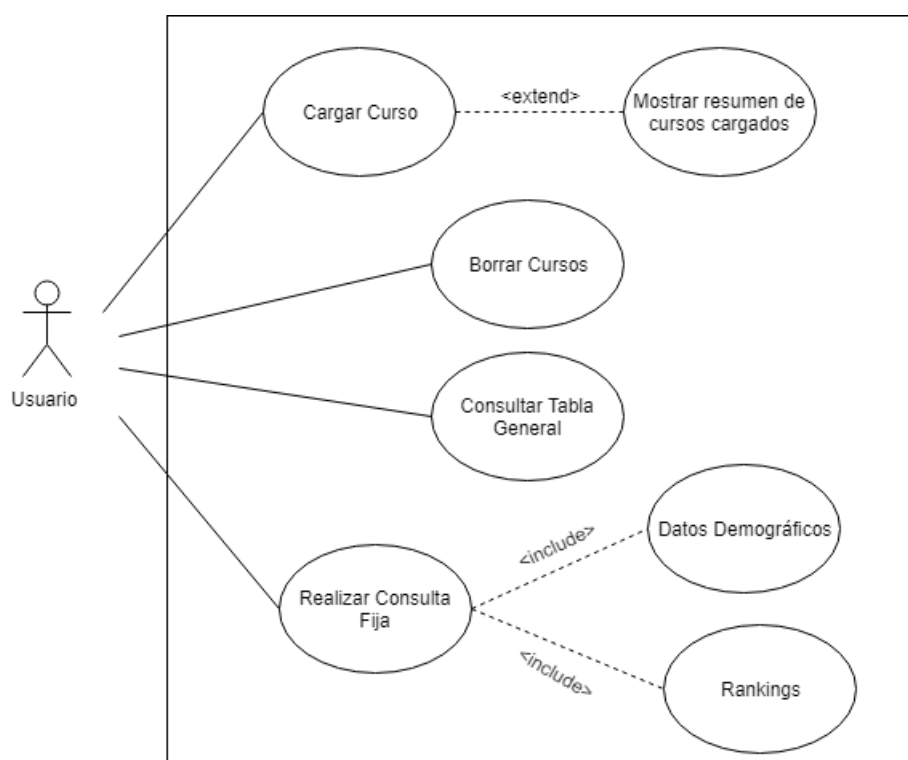
[RNF4] Soporte en Windows: debido a que los ordenadores del grupo de cursos MOOC usa este sistema operativo, la aplicación está adaptada a este sistema operativo.

[RNF5] La aplicación orientada al usuario final soporta el idioma español (castellano).

3.6 Casos de Uso

En la Figura 3, podemos ver el diagrama de casos de uso de la aplicación para el usuario principal de la misma (miembro del grupo de cursos MOOC de la UAM). La parte Análisis de Datos del proyecto no está reflejada debido a que no afecta directamente al usuario final.

Figura 3: Diagrama de Casos de Uso de la aplicación



3.7 Diagrama de la BBDD

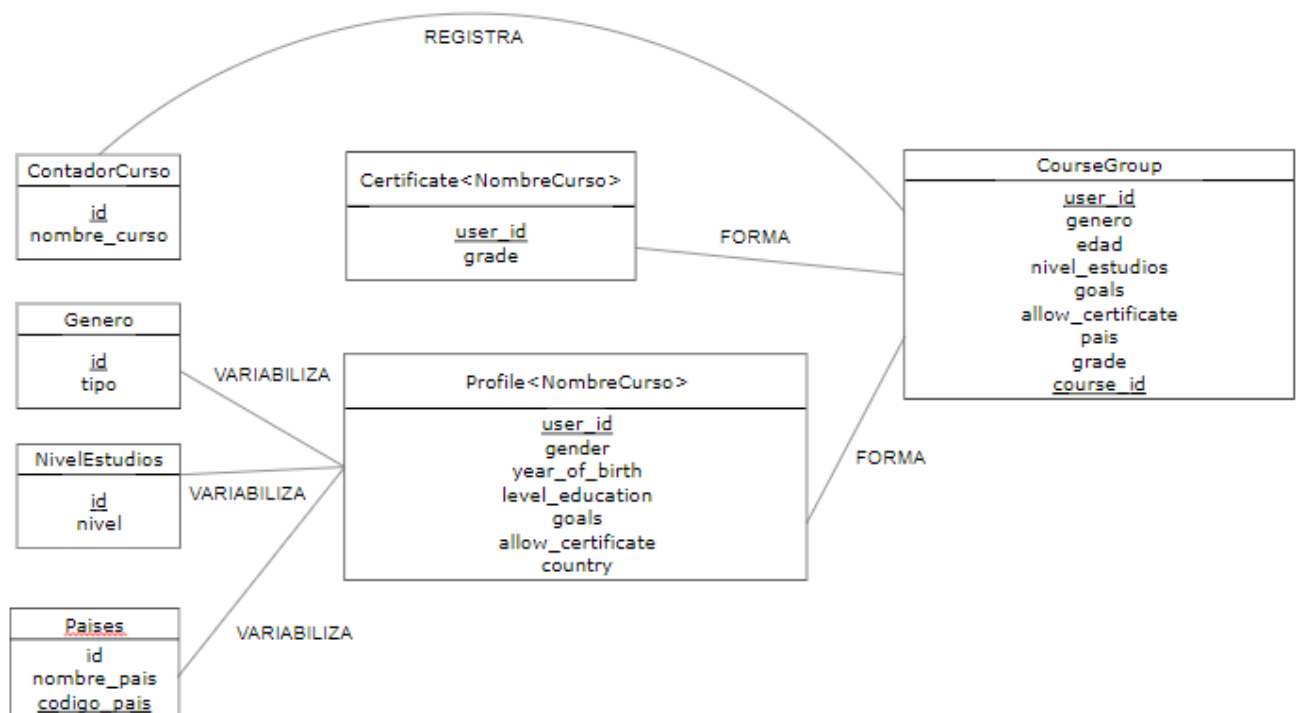


Figura 4: Esquema de la BBDD

La **Figura 4** nos muestra el esquema de la BBDD relacional de la aplicación. Las tablas **Certificate<NombreCurso>** y **Profile<NombreCurso>** (versión resumida con los datos que se necesitan) son generadas a partir de los archivos .csv (Ver apartado 3.4.1 para más información). Algunos campos de la tabla Profile tenían un formato que el cliente quería **reparametrizar** para así volver más uniforme los datos, por tanto, fue necesario crear ciertas **tablas auxiliares**. Dichas tablas contienen la información en el formato correcto⁸.

La tabla **CourseGroup** es la que almacena todos los cursos de la aplicación, y para ello, cada vez que se carga un curso y se generan la tablas **Certificate<NombreCurso>** y **Profile<NombreCurso>**, se realiza un JOIN de ambas para obtener los datos requeridos y en el proceso se sustituyen los parámetros de los campos en formato antiguo, al nuevo especificado por el cliente gracias a dichas tablas auxiliares. Para poder distinguir un curso de otro en **CourseGroup**, es necesario tener un id de cada curso. Dicho id se asigna en la propia carga del curso en otra tabla auxiliar más llamada **ContadorCurso**, en la que se registra el nombre del curso cargado y se registra un id según su orden de carga. Después, en el mismo proceso del JOIN anteriormente mencionado, se mapea el atributo “course_id” con el id del **ContadorCurso**. En el apartado 4.3 se explica con más detalle la estructura de la BBDD.

⁸ El formato de las tablas con sus valores reparametrizados se pueden consultar en el Anexo B.

4 Desarrollo

El desarrollo del proyecto sufrió muchos cambios a lo largo de todo el proceso. En este apartado, aparte de comentar el desarrollo final, comentaremos las etapas previas explicando las decisiones que justifican dichos cambios. Esta sección está dividida en tres subsecciones:

- **Análisis de datos y tratamiento de datos:** antes del desarrollo y programación del proyecto se tuvo que analizar los datos obtenidos previamente para ver qué se podía hacer con ellos y cuales podrían ser de utilidad.
- **Prototipo inicial:** tras la reunión con el cliente se realizó un prototipo funcional del proyecto, tras una revisión se decidieron hacer una serie de cambios para su mejora.
- **Aplicación final:** aplicando los cambios del prototipo, se desarrolló la versión final del proyecto con todos los cambios requeridos.

4.1 Análisis de Datos

Como se ha ido indicando a lo largo del documento, previamente al desarrollo del proyecto, se contaba con los datos de los cursos generados a través de unos scripts. Los datos se distribuían en varios ficheros: desde los logs de la aplicación (archivos donde se encuentra toda la actividad de los cursos: cuantos usuarios han accedido al curso un día, cuantos videos se han reproducido, que dispositivo ha usado el usuario para acceder al curso, etc.) hasta archivos en formato .csv con información de los usuarios de cada curso (demográficos, académicos y sociales).

Para la reunión con el cliente, se listaron todos estos ficheros indicando una breve descripción de qué contenía cada uno para poder hacer una **selección** de los datos. Se decidió que los datos necesarios por el momento eran datos relacionados con la demografía de los estudiantes y sus calificaciones académicas dentro de dichos cursos. Por lo tanto, se decidió usar los archivos Profile.csv y Certificate.csv ya que contenían la información requerida (véase el apartado 3.4.1 RF2 para más detalle).

Dichos datos, y por petición del cliente, tenían que ir en un **formato determinado** por el cliente para poderle ser útil la información. Ciertos campos como el año de nacimiento del usuario debían de cambiar simplemente la edad, y otros como el género, país de origen o nivel de estudios se debían de cambiar a un valor numérico. La asignación de dicho valor se dejó a criterio del desarrollador. Más adelante, en el apartado 4.2, se describirán los criterios escogidos.

En esta primera etapa, el trabajo consistió sobre todo en la lectura de la documentación sobre los datos obtenidos, su entendimiento e interpretación de formatos y tipología. Todo esto sin contar el trabajo previo de documentación para obtener los ficheros con los datos y su tratamiento para poder usarlo en nuestro proyecto.

4.2 Prototipo inicial

Tras la reunión con el cliente, se establecieron los componentes de la aplicación y los procesos a seguir para desarrollar dichos componentes. Los dos componentes principales son: la **web** y la **base de datos**. El componente Web va a ser el encargado de servir de interfaz con el usuario para el uso de la aplicación desarrollada. El componente de Base de Datos es el encargado de gestionar el almacenamiento de los datos para su posterior uso en las consultas SQL del componente web. Debido a la amplitud y sobre todo por los cambios sufridos a lo largo del desarrollo, vamos a centrar la atención primero en el componente de Base de Datos.

Como el objetivo final de la aplicación era realizar consultas sobre un conjunto de datos según las indicaciones del cliente, primero se debía almacenar dicha información en una base de datos. Se decidió **PostgreSQL** como gestor de base de datos debido a la familiaridad del desarrollador con el gestor. Antes de empezar a transformar los datos y darles el formato que quería el cliente, se procedió a hacer simplemente un import de los datos de un curso (ficheros Profile y Certificate de un mismo curso) a modo de prueba. Pare ello, se creó la base de datos “tfg_uamx” para que almacenase los datos de los cursos. Se procedió a hacer el import mediante la instrucción:

```
COPY "TableName" (campo1, campo2, ...) FROM "D:\ruta\del\archivo.csv" WITH  
DELIMITER ';' CSV HEADER;
```

De pronto ocurrió un error en el archivo Profile.csv, indicando que el formato de varias líneas del archivo no era el indicado, debido a que había más columnas que campos definidos. Para entender porque se producía este error debemos comentar la estructura de los archivos csv de UAMx. La primera línea del archivo consiste en los nombres de los campos, separando cada campo con “;”. Es decir, el carácter “;” sirve como **separador** entre una columna y otra. En las siguientes líneas, cada una representa una fila, que contienen los datos relacionados con cada campo, separándose cada campo también con “;”. El foco de error se debía a que el campo “goals” del archivo, recoge la aspiración del estudiante con el curso, dándole libertad de escribir con cualquier carácter. En algunos casos, el estudiante introducía el carácter “;” en su comentario, lo que provocaba a la hora de leer el csv, que se interpretase como que había una columna más en el archivo.

Este error, propició el desarrollo de un script (llamado “validador_archivo.py”) que en principio se encargaría de comprobar el formato del archivo y comprobar que los archivos tuviesen el numero indicado de “;” borrando aquellas del campo “goals” que los usuarios introdujesen al escribir. Dicho script se ejecutaría en la aplicación web en la carga de los cursos. El lenguaje de programación usado fue **Python**, debido a que todos los scripts previos usados para la obtención de datos están en dicho lenguaje, y además al tener que tratar cadenas de texto, se estimó conveniente usar Python.

En un principio, tras este contratiempo, la carga de los cursos en la aplicación se hacía toda **de una sola vez**. Eran necesarios disponer todos los archivos necesarios y a través de un script (“crear_curso.py”) se hacían imports de los archivos, analizándolos y corrigiéndolos antes con “validador_archivo.py”. Este script era un poco rudimentario, porque si se hacía una actualización de los datos (debido a que se han descargado los datos de la semana siguiente), se debía cambiar a mano en el código fuente la dirección de los archivos. Todo esto sin contar con que la carga de los cursos era fija, siempre se cargaban 14 cursos (los

que hay actualmente en la plataforma) no permitiendo así que en un futuro pudiesen cargarse un décimo quinto curso o más cursos.

El proceso de la carga de los cursos sigue el proceso mostrado en la Figura 5:

CARGA DE LOS 14 CURSOS EN BBDD

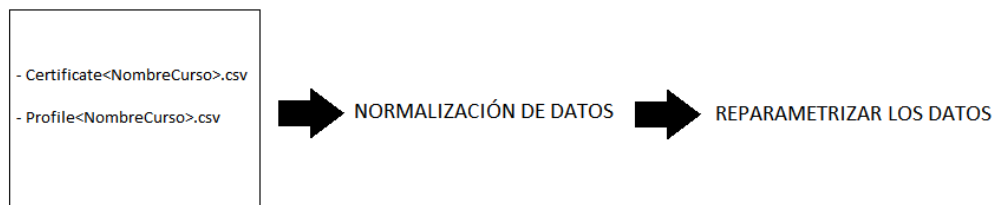


Figura 5: Proceso de carga de los cursos.

La primera etapa, la carga de los 14 cursos, simplemente consta de verificar que los archivos están bien formados y crear **14 tablas Certificate** y **14 tablas Profile**. A continuación, el paso lógico sería reparametrizar los campos indicados directamente, pero al observar los datos en las tablas, vimos algunas incoherencias en los datos. Dichas incoherencias se trataban de algunos campos de una misma columna están vacíos (siendo su valor NULL), pero otras veces están como “null” (siendo la cadena de caracteres, no el valor NULL como antes). Por esto, primero se debe hacer una normalización de los datos, estableciendo todos esos “null” como NULL.

Después de normalizar todos los campos afectados, ya se puede proceder a la reparametrización para así disponer de los datos con el formato requerido por el cliente. El proceso de reparametrizar se realizará cuando se cree la tabla que contenga todos los cursos. A la hora de **insertar los datos**, aquellos que deban ajustarse al nuevo formato, se mapearán con los valores deseados. Los campos a cambiar son: “year_of_birth”, “country”, “educational_level” y “gender”. El primero de esos campos se puede modificar realizando la operación: Año Actual – “year_of_birth”. Para el resto, se crearon tres tablas auxiliares que servirán para mapear los campos. Con esto, asignando un id tipo serial a la tabla auxiliar, conseguimos transformar los valores alfabéticos en numéricos (ejemplo: tabla género el valor masculino pasaría a valer 1 y el valor femenino valdría 2) Para ver toda la estructura y valores de las tablas auxiliares en detalle, puede consultarse el Anexo B.

Como comentábamos anteriormente, el proceso de reparametrizar los datos se hace cuando se crea la tabla con todos los cursos (RF6). Dicha tabla (llamada “CourseGroup”) es la unión de los campos especificados en el RF2. Primero creamos la estructura de la tabla con todos los campos, incluyendo un campo reservado para identificar el curso, y después a través de una sentencia **INSERT** añadimos los datos de las dos tablas. Como hay una correspondencia directa entre Profile (datos de los usuarios) y Certificate (notas de los usuarios) a través de un LEFT JOIN con el user_id de los usuarios se añadió a los usuarios su calificación en el curso (en caso de no haber tenido nota, el campo “grade” se rellenaría con un NULL en la tabla final). Este mismo proceso de usar LEFT JOIN se usó también para reparametrizar los campos especificados. El uso del LEFT JOIN para los campos reparametrizados era, en un principio, solamente un JOIN. Pero al revisar los datos finales, observamos que había una pérdida de usuarios (si en el curso 1 había registrados 600 alumnos, al realizar este proceso el número se podía reducir a 530). Esto se debía a que como había campos con valor NULL, los pasaba por alto y las filas no se añadían a la tabla CourseGroup.

El proceso del INSERT se realizaba 14 veces (una por cada curso) y al final de cada uno se asignaba un id para identificar al curso (del 1 al 14). Siempre se insertaban en el mismo orden por lo tanto el desarrollador sabía en todo momento, según el id, cual era cada curso. De esta forma, quedaría cerrada la parte de la Base de Datos del prototipo inicial.

El **componente Web** hace la función de **interfaz** de usuario. De esta forma se le permite al usuario manipular la aplicación sin tener que acceder al gestor de base de datos y mostrarle así la información de forma amigable y hacer intuitiva el uso de la aplicación. Debido a que el prototipo sirve para mostrar al cliente una especie de demo para ver si el desarrollo de la aplicación se ajusta a las especificaciones del cliente, el componente web no disponía de toda la funcionalidad, solo la básica: cargar todos los cursos, ver la tabla con la información de todos los cursos y tres consultas de ejemplo. Aunque básica, la demo cumplía la funcionalidad para poder hacer una prueba. La estructura del componente se muestra en la Figura 6:

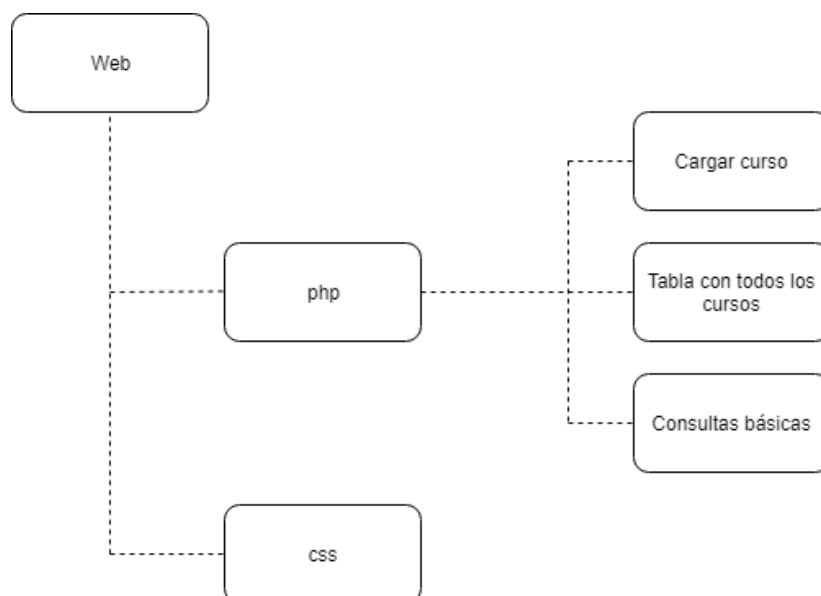


Figura 6: Esquema de la estructura de la Web en el prototipo

Se separó la funcionalidad del diseño, siguiendo las normas de estilo. En la carga del curso, simplemente se ejecutaba el script “crear_curso.py” y se cargaban de golpe los 14 cursos. El formulario php tenía dos campos text, donde el usuario indica donde se encuentran los archivos dentro del sistema (escribir la ruta). En las **consultas básicas**, simplemente eras tres consultas modelo (cuantos usuarios hay en un curso, cuantos han aprobado y quien ha sacado más de una nota determinada) en las cuales, mediante el id del curso, podías hacer la consulta. El problema este modelo de consulta es que requería **conocer de antemano** que id pertenecía a que curso, escribiéndolo a mano en un campo de texto mostrado en la web. Por tanto, la usabilidad de la aplicación no era demasiado alta.

Más adelante, el componente Web sufrió variaciones debido a los cambios que surgieron tras la revisión del prototipo igual que el componente Base de Datos. En el siguiente apartado se verán todas las modificaciones realizadas, explicando por qué y cómo mejoraron la aplicación dichas mejoras. Además, en el apartado 5 se pueden ver imágenes del diseño final de la web.

4.3 Aplicación final

Tras mostrar el prototipo, aunque hubo buenas impresiones, hubo ciertos aspectos que se solicitó una mejora. En primer lugar, el más obvio de todos era hacer más genérica la carga de cursos. Cargar siempre 14 cursos no era práctico, debido a que podía darse la situación de que el cliente solo quisiese estudiar dos cursos, o tres, o incluso más de los 14 de la anterior versión. Además, se quería que fuese el propio usuario quien seleccionase los archivos para su carga en la aplicación y no se tuviese que editar a mano el script

La funcionalidad de borrar los cursos para cargar más se acordó desarrollarla justo ahora. Las consultas estaban bien planteadas, pero le idea de tener que insertar el id a mano sin saber exactamente que id correspondía a cada curso les parecía algo confuso, por lo que se pidió que para seleccionar el curso en la consulta se mostrase como un desplegable con los nombres de los cursos. Se solicitó más consultas dejando en manos del desarrollador que consultas hacer. Se pidió también que la aplicación mostrase **feedback** en caso de error, y que se diese algún mensaje explicativo de que hacer para cargar los cursos o borrar en la parte del componente web.

Para añadir los cursos uno a uno, la estructura de lo planteado anteriormente debe cambiar. Ahora aparte de necesitar los dos archivos .csv, también es necesario el nombre del curso. Como siempre se creaba toda la estructura de tablas y después se añadían los datos en el mismo script, ahora se tenía que controlar el no crear toda la estructura de tablas una y otra vez cada vez que se quiera cargar un curso.

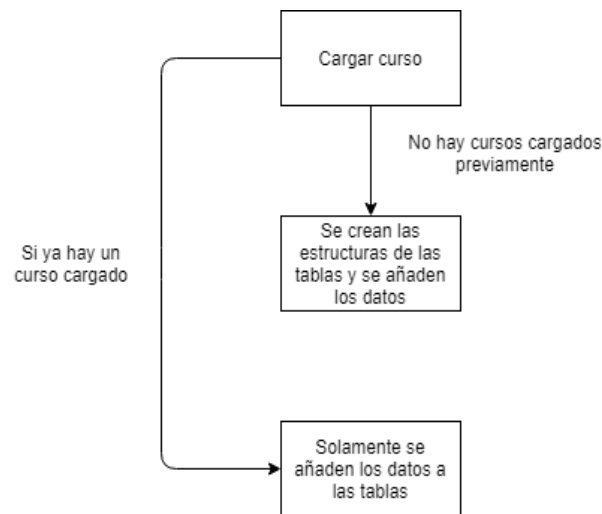


Figura 7: Flujo de la nueva versión de cargar curso

En la Figura 7 podemos observar cómo queda el **nuevo flujo** para la carga del curso. Como en el script se usa sql, simplemente hay que añadir la sentencia IF NOT EXIST a la hora de hacer la creación de tablas. Aun así, hay más cosas que retocar, entre ellas el id de cada curso. Al no hacer los inserts de forma seguida y no saber de forma directa cuantos cursos se han cargado antes, es necesaria la creación de una nueva tabla auxiliar que almacene los ids de los cursos según se van creando. La tabla simplemente contará con dos columnas: id y nombre del curso. Posteriormente, en el proceso de INSERT en la tabla “CourseGroup” se asignará al curso añadido su id correspondiente.

Debido a que ahora se tiene que hacer la carga de los cursos de forma individual, se decidió **unir los scripts** de validación de ficheros y el encargado de crear los cursos en uno solo. De esta forma, se facilita la programación en la parte del componente web.

A partir de aquí el resto de los cambios se produjeron en el componente web, ya que, aparte de realizar las mejoras, se terminaron el resto de las funcionalidades. En el Anexo C se pueden ver los ficheros que componen finalmente la web y una breve explicación de la funcionalidad que abarca cada uno. Lo primero que se hizo fue cambiar el módulo php para la carga de los cursos. Lo primero fue añadir al formulario un campo para escribir el nombre al curso que se va a añadir. Además, se cambiaron los campos text para indicar la ruta de los archivos Profile y Certificate por dos botones que, al clicarlos, desplegasen la ventana de directorios para subir los archivos a la web. En este proceso, también se incluyó una pantalla informativa justo después mostrando un feedback sobre la carga, indicando si todo ha salido bien, o ha ocurrido algún error.

Para facilitar la consulta de cuantos cursos hay cargados en la aplicación y dotar de usabilidad a la aplicación, en la página principal index.php se muestra en forma de tabla una tabla con los cursos cargados o en caso de no haber ninguno, mostrar un mensaje informativo de que todavía no hay ningún curso cargado.

Una vez que se hayan hecho todas las consultas necesarias sobre los cursos almacenados y se quiera analizar nuevos, se deben borrar todos los cursos. Como el volumen de datos no es muy grande en la aplicación usando datos reales, se decidió como método de borrado, simplemente **borrar la base de datos** y crearla de nuevo. De esta forma se simplifica mucho la programación. Debido a que desde una conexión a la base de datos A no se puede borrar esa misma base de datos A, fue necesario crear una base de datos auxiliar (“tfg_auxiliar”) que borrarse la base de datos principal. Todo el proceso de borrado de cursos va acompañado con mensajes informativos para el usuario para saber qué pasos seguir para llevar con éxito la tarea.

Como las consultas restantes se dejaron a juicio del desarrollador, se decidió usar una matriz de decisión (Figura 8) para evaluar qué tipo de consultas pueden ser interesantes, calificando dicho interés como Alto, Medio y Bajo:

	Género	Edad	País	Curso	Calificación	Nivel de Estudios
Género		Medio	Medio	Alto	Bajo	Alto
Edad	Medio		Bajo	Alto	Bajo	Medio
País	Medio	Bajo		Alto	Bajo	Medio
Curso	Alto	Alto	Alto		Bajo	Alto
Calificación	Bajo	Bajo	Bajo	Bajo		Medio
Nivel de Estudios	Alto	Medio	Medio	Alto	Medio	

Figura 8: Matriz de Decisión

Tras hacer esta primera selección, y eligiendo las consultas con un nivel de interés Alto, se pasó la lista al cliente, dando el visto bueno a las consultas elegidas. Al analizar la matriz, al ver que el campo Curso se repetía constantemente, se decidió incluirlo en el resto de las consultas como parámetro, pudiendo hacer la consulta sobre un curso en concreto o sobre todos los cursos existentes. El motivo por el cual no se le da demasiada importancia al campo Calificación, se debe a que en muchos cursos no hay demasiadas calificaciones o el nivel de participación no es muy alto, aun así (para los casos en los que sí que haya participación) se desarrolló una consulta para obtener a los alumnos que han sacado más de una nota determinada (ya sea en un curso o en todos los cursos). La lista de consultas creadas es la siguiente:

- Mostrar la tabla general de todos los cursos con sus alumnos.
- Usuarios según su edad que estudian en un curso determinado (o en todos).
- Usuarios que han sacado más de una nota determinada en un solo curso o en todos los cursos.
- Consulta sobre la cantidad de hombres y mujeres estudian en un curso o en todos los cursos.
- Determinando el género, ver su nivel de estudios (titulación, máster, estudios básicos, etc.) en un curso determinado (o en todos).
- Cuál es el nivel de estudios de un curso (sin hacer distinción por género).
- Obtener las nacionalidades de los alumnos que estudian un curso determinado.
- Determinando género y curso (pudiendo ser sobre uno o todos), ver cuántos usuarios por país estudian.

Cada una de estas consultas se desarrolla en un **fichero php** distinto haciendo su correspondiente conexión a base de datos y realizando la consulta sql, para añadir más consultas para un futuro, simplemente sería necesario crear un nuevo fichero php dentro del servidor XAMPP con la consulta y después listarlo en el fichero php donde vienen todos los links a las páginas de consulta.

Se estimó conveniente también añadir consultas para obtener información en formato de ranking, para así obtener cierta información de forma más directa. Dichas consultas eran: cursos con más alumnos y los países que más alumnos tienen registrados en los cursos. Como se dijo anteriormente y siguiendo el mismo proceso, se pueden añadir más consultas.

5 Integración, pruebas y resultados

5.1 Integración

A continuación, se mostrarán capturas de pantalla de las diferentes funcionalidades ya integradas en la aplicación explicando, cuando sea necesario, la captura de la aplicación.

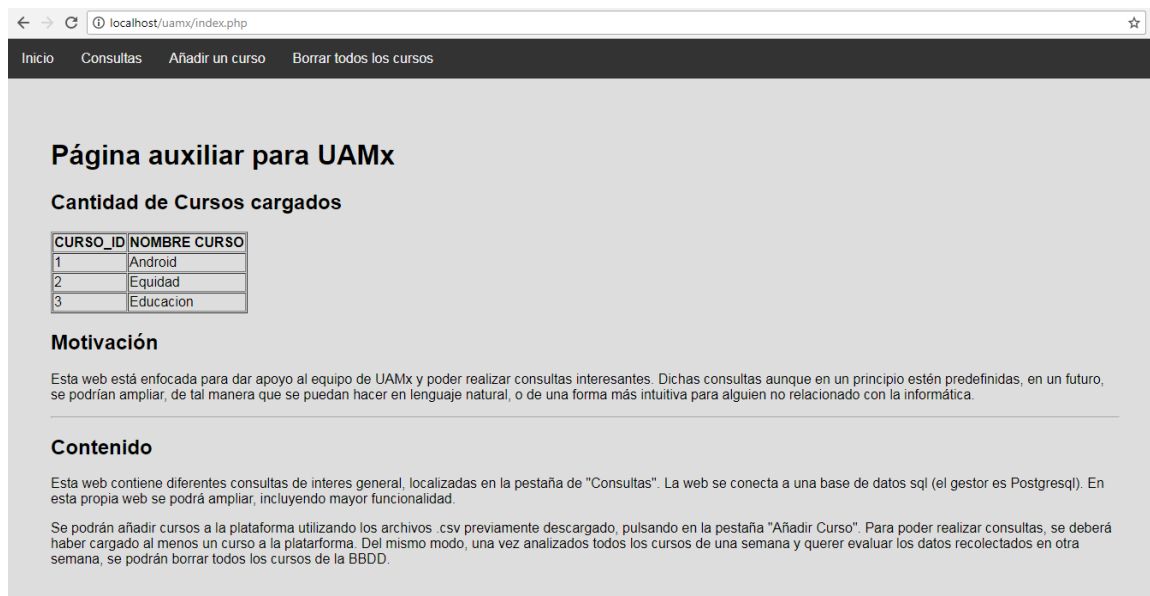


Figura 9: Página de inicio de la aplicación.

En la Figura 9 se muestra la página de inicio de la aplicación web. En la parte superior aparecen los diferentes enlaces (en formato toolbar) a las diferentes funcionalidades (los nombres son bastante explicativos). En esta página de inicio también se muestran la tabla con los cursos cargados y la **motivación y contenido** de la web. Vemos que hay ya cargados una serie de cursos, por lo que vamos a proceder a borrarlos y cargar unos nuevos. En la Figura 10 y 11 se muestra cómo.

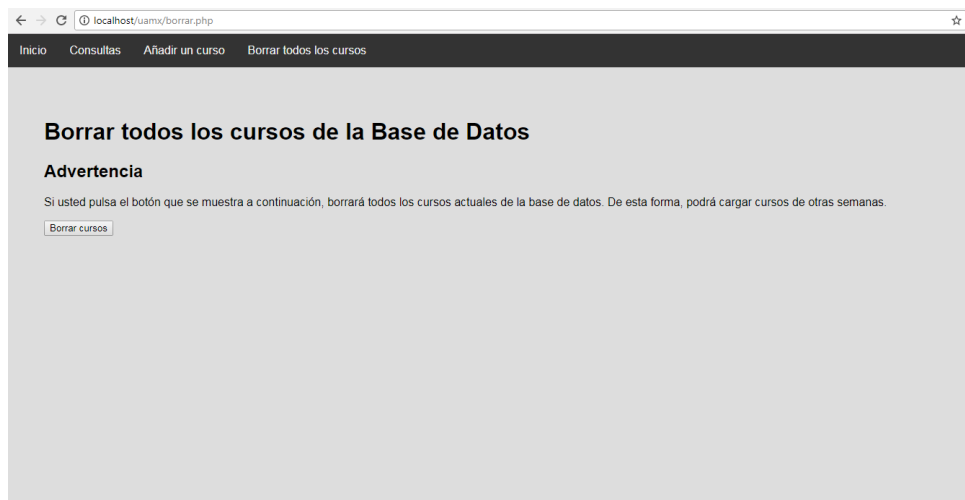


Figura 10: Primera parte del proceso de borrar los cursos.

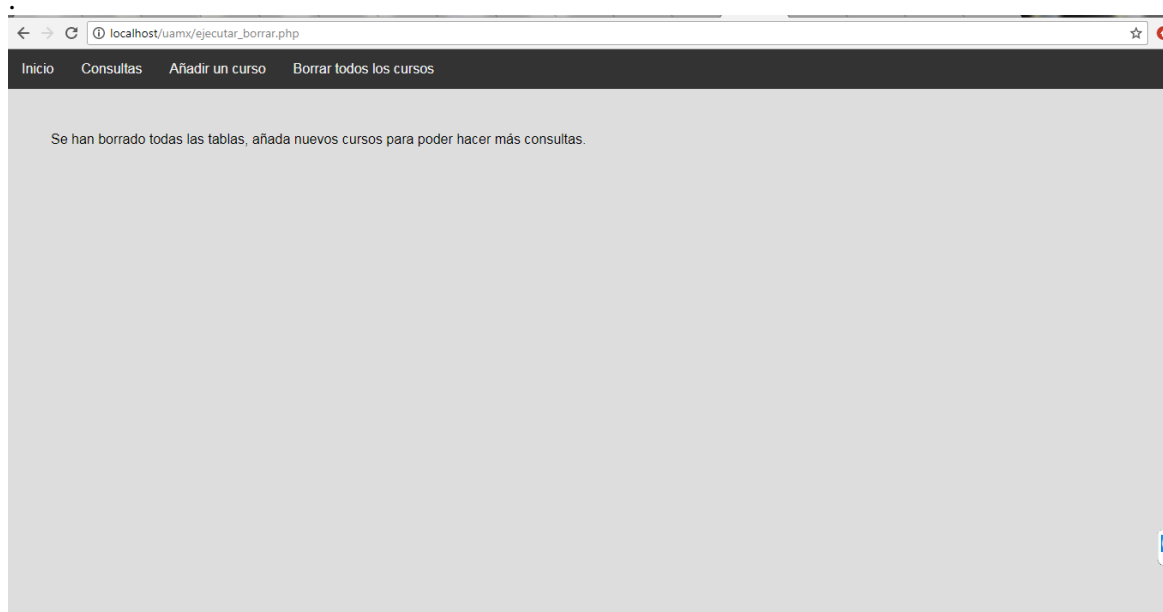


Figura 11: Mensaje informativo al terminar el proceso de borrar los cursos.

Si volviésemos a la página de inicio, donde antes se mostraba la tabla de los cursos cargados, no se mostraría nada, solo un mensaje: “Aún no se ha cargado ningún curso”. Ahora, vamos a añadir dos cursos a la plataforma. En la Figura 12 se muestra un formulario para la carga del curso y en la Figura 13 se muestra el **mensaje informativo** de éxito o error en la carga de dicho curso.

A screenshot of a web browser window. The address bar shows 'localhost/uamx/anadir.php'. The navigation bar has links: 'Inicio', 'Consultas', 'Añadir un curso', and 'Borrar todos los cursos'. The main content area has the title 'Añadir un curso a la Base de Datos' and the instruction 'Inserte los campos especificados a continuación'. Below this, there is a text input field for 'Nombre del curso:'. Then, two file selection fields: 'Localización Profile:' and 'Localización Certificate:', each with a 'Seleccionar archivo' button and the text 'Ningún archivo seleccionado'. At the bottom of the form is an 'Enviar' button.

Figura 12: Pantalla para cargar un curso.

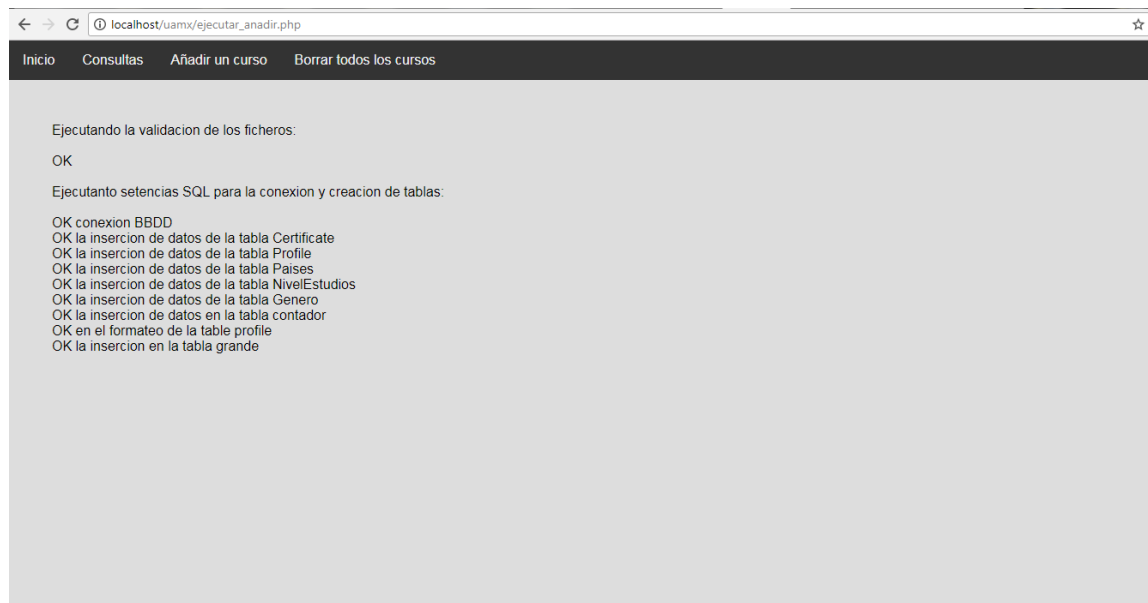


Figura 13: Mensaje informativo de la carga de cursos.

A continuación, para comprobar la funcionalidad de consultas, se cargaron dos cursos (llamados Equidad y Android) para poder realizar las consultas. Para acceder al menú de consultas, simplemente en la parte superior se debería dar clic en “Consultas” y se mostraría una pantalla con el contenido de la Figura 14.



Figura 14: Menú de consultas

Cada uno de los enlaces redirige a un tipo de consulta diferente. El primero como se muestra en la figura 15, obtendrá la tabla con todos los cursos cargados (RF6), el segundo enlace redirigirá a la página de consultas de tipo demográfico en la que se podrán realizar diferentes tipos de consultas (Figura 16) y en el tercero se mostrarán los diferentes tipos de consultas tipo ranking que hay.

← → ↻ localhost/uamx/tabla_cursos.php ☆

Tabla de usuarios de todos los cursos

USER_ID	GENERO	EDAD	NIVEL_ESTUDIOS	OBJETIVOS	ALLOW_CERTIFICATE	PAIS	CALIFICACION	COURSE_ID
4135324	1	26	2		1	2		1
16193279	1	32	2		1	10		1
16127573	1	55	2		1	10		1
16110549	1	49	2		1	10		1
6488066	1	43	2	aprender	1	10		1
16603877	2	53	1	Por trabajar desde hace 28 años con Discapacidad psicomotriz en educac pública Especialmente por haberme desempeñado en Gabinete de informática educativo didáctica 13 años.	1	10		1
16550991	2	22	1	Cursos mooc	1	10		1
16507243	2	33	1		1	10		1
16151977	2	50	1	Por las capacitaciones que propone	1	10		1
16145938	2	28	1	PROFUNDIZAR MIS CONOCIMIENTOS EN PROCESOS COMO INCLUSIÓN ESCOLAR YA QUE SOY PROFESORA EN EDUCACIÓN ESPECIAL CON ORIENTACIÓN EN DISCAPACIDAD INTELECTUAL	1	10		1
16111325	2	35	1	Para poder incorporar herramientas	1	10		1
16109570	2	56	1	Como asesora Técnica de la Dirección Provincial del Ministerio de Educación de la Provincia de Santa Fe de Argentina, se esta trabajando en relación a la Inclusión atendiendo a la Convención de los derechos de personas con discapacidad, con la ley nacional de Educación y las normativas provinciales. En el transcurso de este año se han realizado diferentes convenios de inclusión con organismos nacionales e internacionales. El derecho a que todos los niños/as, jóvenes y adultos puedan realizar sus trayectorias educativas atendiendo a la accesibilidad no solo en relación a los ajustes razonables	1	10		1

Figura 15: Fragmento de la tabla global de todos los cursos.

← → ↻ localhost/uamx/consultas_demograficas.php ☆

CONSULTAS TÉCNICAS SOBRE LOS CURSOS DE UAMx

Datos de los alumnos de un curso y edad (mayor o igual a esa edad):

Edad:

Curso:

Alumnos que han sacado más de una nota determinada en un curso determinado:

Nota:

Curso:

Cantidad de alumnos por país que hay en cada curso

País:

Curso:

Figura 16: Fragmento de las consultas demográficas.

En la Figura 16 se puede ver que en el campo curso, presente en la gran mayoría de las consultas, el campo es un desplegable. En el desplegable se encuentran las opciones de cada curso cargado, mostrando los nombres de todos los cursos o “Todos” en cuyo caso las consultas se harán sobre todos los cursos. En la Figura 17 se muestra el resultado de una consulta sobre el nivel de estudios de un curso (o todos) y por último en la Figura 18 se muestra el resultado de una consulta de tipo ranking.

Nivel de estudios del curso 1

NIVEL	USUARIOS
1	168
2	526
3	3
4	92
5	33
6	310
7	2
8	34
9	39
	226

Volver a la página principal

Figura 17: Ejemplo de consulta demográfica

Ranking de los paises que se inscriben a cursos UAMx

PAIS	CANTIDAD
142	1037
202	807
173	345
61	323
47	309
	278
10	270
43	163
236	86
232	83

Volver a la página principal

Figura 18: Ejemplo de consulta tipo ranking

5.2 Pruebas y resultados

Las pruebas realizadas para comprobar el correcto funcionamiento de la aplicación fueron realizadas en el equipo proporcionado por el cliente. Se realizó una prueba de estrés a través de un script que carga **100 cursos** en la aplicación. Con esta prueba podremos observar cuanto tiempo tardaría en procesarlos adecuadamente y cargarlos. Además, una vez cargados los 100 cursos, estudiaremos cuanto tiempo tardaría en hacer las consultas y cuánto tardaría el sistema en borrar los cursos.

El script encargado de la prueba de estrés está desarrollado en Python y es una rutina que ejecuta un bucle tantas veces como cursos se quieran cargar. En la Figura 19 se muestra dicho script y como se ha realizado el cálculo de los tiempos.

```

1  #!/usr/bin/python2.7
2
3  import sys
4  import os
5  import time
6
7  if __name__ == '__main__':
8      print "Ejecutando prueba de estres:\n"
9      print "Generacion de 100 cursos\n"
10     var = "D:\\Czar\\csv-data\\ProfileAndroid301x3T2017.csv"
11     var2 = "D:\\Czar\\csv-data\\CertificateAndroid301x3T2017.csv"
12
13     inicio = time.clock()
14     for i in range (100):
15         ejec = "python D:\\usuarios\\UAMx\\Desktop\\TFG\\crear_curso.py " + str(i) + " " + var + " " + var2
16         os.system(ejec)
17     fin = time.clock()
18     resultado = fin + inicio
19     print "Prueba Finalizada. El tiempo transcurrido ha sido de " + str(resultado) + " segundos"

```

Figura 19: Script de prueba de estrés.

Se ha decidido usar para dicha prueba, el conjunto de datos perteneciente al curso de Android que oferta UAMx, ya que es de los cursos con mayor cantidad de alumnos. Este curso en concreto cuenta con **2807 registros** Con esta prueba se consigue someter a la aplicación a un estrés de carga de 100 cursos con un volumen de datos elevado. En la Figura 20 se puede ver el resultado tras la ejecución del script.

```

C:\Windows\system32\cmd.exe
OK conexion BBDD
OK la insercion de datos de la tabla Certificate
OK la insercion de datos de la tabla Profile
OK la insercion de datos de la tabla Paisier
OK la insercion de datos de la tabla NivelEstudios
OK la insercion de datos de la tabla Genero
OK la insercion de datos en la tabla contador
OK en el formato de la table profile
OK la insercion en la tabla grande
Ejecutando la validacion de los ficheros:
OK
Ejecutando setencias SQL para la conexion y creacion de tablas:
OK conexion BBDD
OK la insercion de datos de la tabla Certificate
OK la insercion de datos de la tabla Profile
OK la insercion de datos de la tabla Paisier
OK la insercion de datos de la tabla NivelEstudios
OK la insercion de datos de la tabla Genero
OK la insercion de datos en la tabla contador
OK en el formato de la table profile
OK la insercion en la tabla grande
Ejecutando la validacion de los ficheros:
OK
Ejecutando setencias SQL para la conexion y creacion de tablas:
OK conexion BBDD
OK la insercion de datos de la tabla Certificate
OK la insercion de datos de la tabla Profile
OK la insercion de datos de la tabla Paisier
OK la insercion de datos de la tabla NivelEstudios
OK la insercion de datos de la tabla Genero
OK la insercion de datos en la tabla contador
OK en el formato de la table profile
OK la insercion en la tabla grande
Ejecutando la validacion de los ficheros:
OK
Ejecutando setencias SQL para la conexion y creacion de tablas:
OK conexion BBDD
OK la insercion de datos de la tabla Certificate
OK la insercion de datos de la tabla Profile
OK la insercion de datos de la tabla Paisier
OK la insercion de datos de la tabla NivelEstudios
OK la insercion de datos de la tabla Genero
OK la insercion de datos en la tabla contador
OK en el formato de la table profile
OK la insercion en la tabla grande
Ejecutando la validacion de los ficheros:
OK
Prueba Finalizada. El tiempo transcurrido ha sido de 49.7787957498 segundos
D:\usuarios\UAMx\Desktop\TFG>

```

Figura 20: Resultados de la carga de 100 cursos.

El tiempo total empleado es de 49,78 segundos (redondeando, 50 segundos). El caso de prueba se ha realizado en las mismas situaciones que con la carga de un curso normal usando el mismo equipo que se ha utilizado durante todo el desarrollo (es decir el equipo que nos proporcionó la UTED). Al ser la misma situación, también se muestran todos los mensajes de feedback de la aplicación, lo que provoca una ralentización al mostrar tantos mensajes. Tras consultar la base de datos, la tabla “CourseGroup” tiene 280700 registros al finalizar la ejecución de la prueba. Son, por tanto, 50 segundos para 100 cursos, de lo que se deduce que se cargan 2 cursos con un gran volumen de datos cada segundo, lo que es un tiempo aceptable ya que cumple sobradamente con el RNF3. Se estima por tanto que se ha pasado con éxito esta prueba el sistema.

Para el estudio de los tiempos de ejecución de las consultas, se usó el propio gestor de bases de datos para realizar dichas consultas. Se hizo así debido a que intentando usar otras métricas sobre la web, los tiempos de ejecución eran tan pequeños que muchas veces se obtenía un tiempo de 0 segundos. Las consultas se hicieron sobre los 100 cursos cargados previamente y se realizó la prueba sobre tiempos de ejecución en dos tipos de consultas diferentes: una demográfica y otra de rankings. Ambas se hicieron sobre el total de datos y no sobre uno solo, ya que así se dota de mayor complejidad y cantidad de datos. En las Figuras 21 y 22 están los dos ejemplos usados.

```
tfg_uamx on postgres@PostgreSQL 10
1 EXPLAIN analyze
2 SELECT nivel_estudios, count(*)
3 FROM public."CourseGroup" GROUP BY nivel_estudios ORDER BY nivel_estudios ASC
```

Figura 21: Estudio sobre el nivel de estudios general de todos los cursos.

```
tfg_uamx on postgres@PostgreSQL 10
1 EXPLAIN ANALYZE
2 SELECT course_id, count(*) AS usuarios
3 FROM public."CourseGroup" GROUP BY course_id ORDER BY usuarios DESC
```

Figura 22: Estudio sobre el ranking de los cursos más estudiados.

Gracias a la sentencia **EXPLAIN ANALYZE** podemos obtener información relevante sobre la ejecución de dichas consultas sobre la base de datos. Aunque solo nos interese el tiempo de ejecución en este caso, los Query Plans generados a través de la instrucción EXPLAIN ANALYZE puede ayudarnos a mejorar el rendimiento de nuestras consultas ya que ofrece información como el tipo de algoritmo usado, coste de lo consulta, cuantos bucles ha tenido que ejecutar, la clave que ha usado, etc. En las Figuras 23 y 24 se muestran dichas Query Plans y comentaremos los resultados obtenidos.

Figura 23: Query Plan de la 1ª consulta

QUERY PLAN	text
1	Finalize GroupAggregate (cost=6472.08..6472.31 rows=9 width=12) (actual time=4...
2	Group Key: nivel_estudios
3	-> Sort (cost=6472.08..6472.13 rows=18 width=12) (actual time=44.458..44.459 row...
4	Sort Key: nivel_estudios
5	Sort Method: quicksort Memory: 26kB
6	-> Gather (cost=6469.82..6471.71 rows=18 width=12) (actual time=44.426..44.444 r...
7	Workers Planned: 2
8	Workers Launched: 2
9	-> Partial HashAggregate (cost=5469.82..5469.91 rows=9 width=12) (actual time=2...
10	Group Key: nivel_estudios
11	-> Parallel Seq Scan on "CourseGroup" (cost=0.00..4884.88 rows=116988 width=4) ...
12	Planning time: 0.066 ms
13	Execution time: 47.066 ms

QUERY PLAN	text
1	Sort (cost=6503.91..6504.16 rows=99 width=12) (actual time=52.463..52.466 rows=100 lo...
2	Sort Key: (count(*)) DESC
3	Sort Method: quicksort Memory: 29kB
4	-> Finalize GroupAggregate (cost=6498.16..6500.63 rows=99 width=12) (actual time=52.4...
5	Group Key: course_id
6	-> Sort (cost=6498.16..6498.65 rows=198 width=12) (actual time=52.398..52.404 rows=20...
7	Sort Key: course_id
8	Sort Method: quicksort Memory: 34kB
9	-> Gather (cost=6469.82..6490.61 rows=198 width=12) (actual time=52.296..52.342 rows=...
10	Workers Planned: 2
11	Workers Launched: 2
12	-> Partial HashAggregate (cost=5469.82..5470.81 rows=99 width=12) (actual time=25.888...
13	Group Key: course_id
14	-> Parallel Seq Scan on "CourseGroup" (cost=0.00..4884.88 rows=116988 width=4) (actua...
15	Planning time: 1.036 ms
16	Execution time: 55.651 ms

Figura 24: Query Plan de la 2ª consulta

El tiempo de ejecución es (redondeando) en la primera 47 ms y en la segunda 55 ms. Traducido al usuario final sería que dichas consultas, aún con 100 cursos cargados, se hacen en un tiempo despreciable para el usuario. Se puede ver que el método de ordenación ha sido Quicksort en ambos casos, esto se debe a que se tenían que ordenar los resultados de dichos datos de forma ascendente y descendente respectivamente, por lo que este algoritmo es el que mejor.

El tiempo empleado en el borrado de cursos es el mismo que antes, debido a que se borra la base de datos y se crea una de nuevo.

6 Conclusiones y trabajo futuro

6.1 Conclusiones

Desde el inicio del desarrollo del proyecto; incluso antes, en los días de reunión con el tutor, se seleccionó este Trabajo de Fin de Grado debido a las áreas de la informática que se desarrollaban. El análisis de datos es una disciplina puntera hoy en día en la informática y tener un primer acercamiento (aunque en menor medida y de forma más liviana) fue uno de los alicientes de la selección del proyecto. La integración de esto con una aplicación web fue la excusa perfecta para aventurarme a elegir este trabajo.

Durante el desarrollo fui observando cómo se alzaban ante mí diferentes problemas de todo tipo, que día a día los desarrolladores de software tienen que resolver problemas como estos o incluso peores. Han sido experiencias que han enriquecido mi aprendizaje y, sobre todo, han ampliado mi visión sobre lo que es trabajar en un proyecto de verdad con un cliente con el que reunirse y comunicarte para obtener feedback por ambas partes del desarrollo del proyecto.

El proyecto partió de una serie de precondiciones y datos por parte del cliente, y la aplicación desarrollada fue el puente entre el estado previo y la funcionalidad que ellos querían gracias a los componentes elaborados. Del proceso de desarrollo de la base de datos se ha aprendido a seguir un proceso en cadena pasando por diferentes etapas hasta tener los datos ofrecidos para el estudio en el formato adecuado; teniendo que además variar la estructura de la base de datos en el proceso para mejorar el sistema y así dotar de mayor usabilidad a la aplicación. En el proceso de desarrollo de la web, a pesar de que a lo largo de la carrera universitaria se han realizado tareas similares en determinadas asignaturas, aquí no había ningún tipo de guía y todas las decisiones de diseño y utilidad debían de ser acordes a lo solicitado por el cliente. Aun siendo un proceso “familiar” con el desarrollador, se han aprendido cosas nuevas sobre la carga de ficheros en una web, obtención de datos a través de consultas y mostrar el resultado en un formato legible para el usuario.

Aunque no era el objetivo del TFG, el hecho de tratar con un cliente y tener que aplicar también (hasta cierto punto) lo aprendido sobre ingeniería del software, dota al proyecto de mayor profundidad, ya que no es simplemente una aplicación; es un proyecto como tal, sin ciertas exigencias como tener fechas de entrega (no se considera la fecha de entrega del TFG ya que es un periodo bastante extenso e indefinido), o tener que organizarse un equipo y por tanto utilizar alguna metodología de desarrollo (por ejemplo una metodología ágil como SCRUM).

Valoro positivamente el trabajo realizado, tanto por mi parte como por mi tutor que ha sabido guiarme por las diferentes etapas del proyecto.

6.2 Trabajo futuro

Debido a la reciente creación de la aplicación hay muchas mejoras que se podrían aplicar al sistema. A continuación, se enumeran algunas labores de mejora que se podrían realizar con una breve explicación:

- Añadir seguridad: debido a que la aplicación está enfocada a un uso local del grupo UTED y no tendría sentido que se atacasen a sí mismos (en un escenario común), no se añadió seguridad al sistema de consultas, por lo que se podría hacer inyección sql y supondría un riesgo a la integridad de la aplicación.
- Consultas dinámicas: otro foco de trabajo sería sustituir el sistema de consultas fijas que hay actualmente por un sistema de consulta dinámica a través del lenguaje natural. Esto podría permitir un aumento exponencial de la cantidad de consultas que se pudiesen hacer sobre el sistema.
- Almacenamiento semanal: podría llegar a ser de interés hacer que en vez de borrar los cursos de una semana para cargar los de una nueva, añadir una funcionalidad para almacenar cursos antiguos, de esta forma se podrían comparar en la propia aplicación los datos de dos momentos temporales diferentes. Incluso se podrían añadir procesos batch al servidor para realizar la descarga y anonimización de datos semanalmente sin que nadie tuviese que hacerlo a mano.
- Mejorar la interfaz gráfica: debido a que el objetivo del TFG no era hacer una web “bonita” sino funcional y que cumpliese los requisitos del cliente, no se hizo demasiado hincapié en este apartado (se hizo lo necesario para cumplir con unos requisitos mínimos, nada más).

Referencias

- [1] Dave Cormier (2012), “What is a MOOC?”
<https://www.youtube.com/watch?v=eW3gMGqcZQc>
- [2] SIEMENS, G. (2012), “What is the theory that underpins our moocs?”,
<http://www.elearnspace.org/blog/2012/06/03/what-is-the-theory-that-underpins-our-moocs/>
- [3] Tony Bates (October 2014), “Comparing xMOOCs and cMOOCs: philosophy and practice”, <https://www.tonybates.ca/2014/10/13/comparing-xmoocs-and-cmoocs-philosophy-and-practice/>
- [4] Zaid Ali Alsagoff, “Demystifying MOOCs!”,
<https://www.slideshare.net/zaid/demystifying-moocs/> , pp. 13.
- [5] Esteban Vázquez-Cano, Eloy López Meneses, “La filosofía educativa de los MOOC y la educación universitaria” en Ried, Vol. 18, núm. 2 (2015), pp. 27.

Glosario

MOOC *Massive Open Online Course*

Cursos en línea dirigidos a un número ilimitado de participantes a través de Internet según el principio de educación abierta y masiva.

BBDD *Bases de Datos*

Conjuntos de datos que pertenecen a un mismo entorno o contexto, almacenados de forma sistemática para su uso futuro.

PHP *PHP: Hypertext Preprocessor*

Lenguaje de programación de propósito general de código del lado del servidor. Diseñado para el desarrollo web de contenido dinámico.

SQL *Structured Query Language*

Lenguaje estandarizado para la definición y consulta de bases de datos relacionales.

PYTHON

Lenguaje de programación interpretado, multiparadigma, tipado dinámico y multiplataforma.

XAMPP

Servidor web de plataforma, software libre. Consiste en el sistema de gestión de bases de datos MySQL, el servidor web Apache y los intérpretes para lenguajes de script PHP y Perl

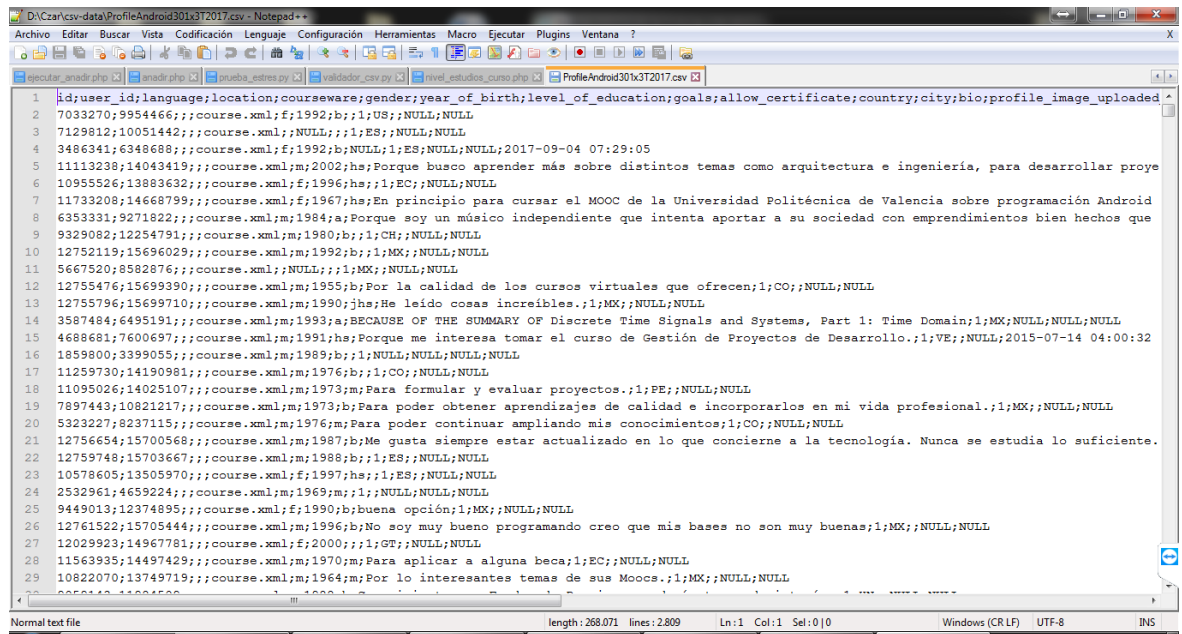
PostgreSQL

Sistema de gestión de bases de datos relacional orientado a objetos y libre.

Anexos

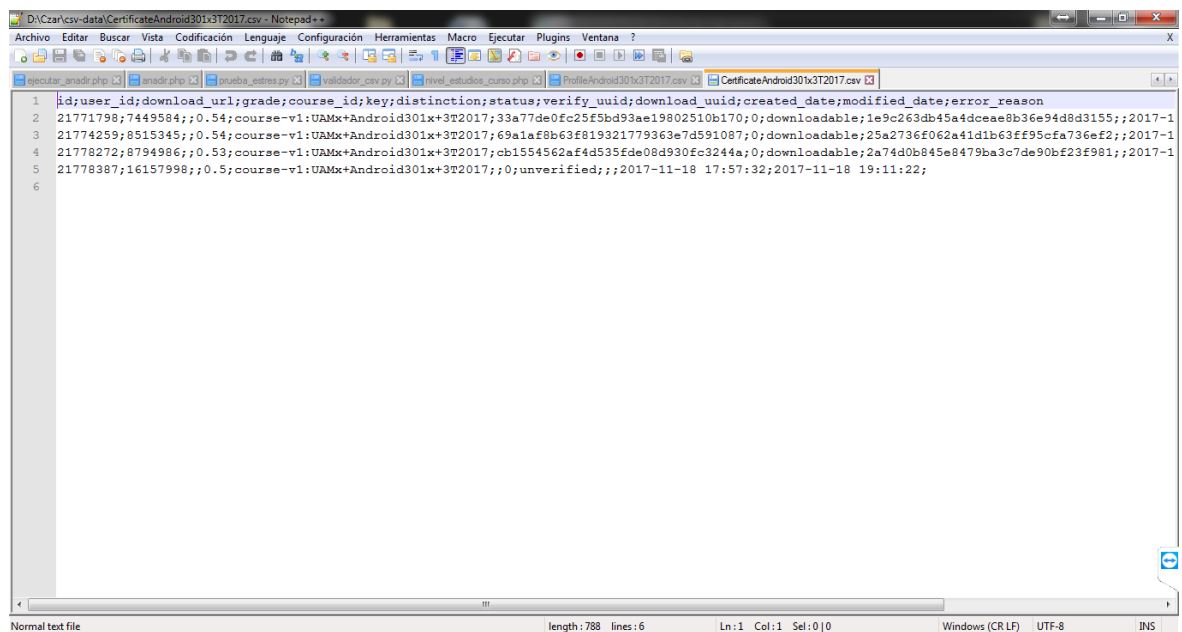
A Formato de los ficheros Certificate y Profile

A continuación, se muestran dos ejemplos (uno por cada fichero) de cómo es un fichero Certificate.csv y Profile.csv.



```
1 id;user_id;language;location;courseware;gender;year_of_birth;level_of_education;goals;allow_certificate;country;city;bio;profile_image_uploaded
2 7033270;9954466;;;course.xml;f;1992;b;;1;US;;NULL;NULL
3 7129812;10051442;;;course.xml;f;1992;b;NULL;1;ES;NULL;NULL
4 3486341;6348688;;;course.xml;f;1992;b;NULL;1;ES;NULL;NULL;2017-09-04 07:29:05
5 11113238;14043419;;;course.xml;m;2002;hs;Porque busco aprender más sobre distintos temas como arquitectura e ingeniería, para desarrollar proye
6 10955526;13883632;;;course.xml;f;1996;hs;;1;EC;;NULL;NULL
7 11733208;14668799;;;course.xml;f;1967;hs;En principio para cursar el MOOC de la Universidad Politécnica de Valencia sobre programación Android
8 6353331;9271822;;;course.xml;m;1984;a;Porque soy un músico independiente que intenta aportar a su sociedad con emprendimientos bien hechos que
9 9329082;12254791;;;course.xml;m;1980;b;;1;CH;;NULL;NULL
10 12752119;15696029;;;course.xml;m;1992;b;;1;MX;;NULL;NULL
11 5667520;8582876;;;course.xml;f;1992;b;;1;MX;;NULL;NULL
12 12755476;15699390;;;course.xml;m;1955;b;Por la calidad de los cursos virtuales que ofrecen;1;CO;;NULL;NULL
13 12755796;15699710;;;course.xml;m;1990;jhs;He leído cosas increíbles.;1;MX;;NULL;NULL
14 3587484;6495191;;;course.xml;m;1993;a;BECAUSE OF THE SUMMARY OF Discrete Time Signals and Systems, Part 1: Time Domain;1;MX;NULL;NULL;NULL
15 4688601;7600697;;;course.xml;m;1991;hs;Porque me interesa tomar el curso de Gestión de Proyectos de Desarrollo.;1;VE;;NULL;2015-07-14 04:00:32
16 1859800;3399055;;;course.xml;m;1989;b;;1;NULL;NULL;NULL;NULL
17 11259730;14190981;;;course.xml;m;1976;b;;1;CO;;NULL;NULL
18 11095026;14025107;;;course.xml;m;1973;m;Para formular y evaluar proyectos.;1;PE;;NULL;NULL
19 7897443;10821217;;;course.xml;m;1973;b;Para poder obtener aprendizajes de calidad e incorporarlos en mi vida profesional.;1;MX;;NULL;NULL
20 5323227;8237115;;;course.xml;m;1976;m;Para poder continuar ampliando mis conocimientos;1;CO;;NULL;NULL
21 12756654;15700568;;;course.xml;m;1987;b;Me gusta siempre estar actualizado en lo que concierne a la tecnología. Nunca se estudia lo suficiente.
22 12759748;15703667;;;course.xml;m;1988;b;;1;ES;;NULL;NULL
23 10578605;13505970;;;course.xml;f;1997;hs;;1;ES;;NULL;NULL
24 2532961;4659224;;;course.xml;m;1969;m;;1;NULL;NULL;NULL
25 9449013;12374895;;;course.xml;f;1990;b;buena opción;1;MX;;NULL;NULL
26 12761522;15705444;;;course.xml;m;1996;b;No soy muy bueno programando creo que mis bases no son muy buenas;1;MX;;NULL;NULL
27 12029923;14967781;;;course.xml;f;2000;;;1;GT;;NULL;NULL
28 11563935;14497429;;;course.xml;m;1970;m;Para aplicar a alguna beca;1;EC;;NULL;NULL
29 10822070;13749719;;;course.xml;m;1964;m;Por lo interesantes temas de sus Moocs.;1;MX;;NULL;NULL
```

Figura 25 de Anexo A: Fichero Profile.csv del curso Android.



```
1 id;user_id;download_url;grade;course_id;key;distinction;status;verify_uuid;download_uuid;created_date;modified_date;error_reason
2 21771798;7449584;;0.54;course-v1:UAMx+Android301x+3T2017;33a77de0fc25f5bd93ae19802510b170;0;downloadable;1e9c263db45a4dcae8b36e94d8d3155;;2017-1
3 21774259;8515345;;0.54;course-v1:UAMx+Android301x+3T2017;69a1af8b63f819321779363e7d591087;0;downloadable;25a2736f062a41db63ff95cfa736ef2;;2017-1
4 21778272;8794986;;0.53;course-v1:UAMx+Android301x+3T2017;cb1554562af4d535fde08d930fc3244a;0;downloadable;2a74d0b845e8479ba3c7de90bf23f981;;2017-1
5 21778387;16157998;;0.5;course-v1:UAMx+Android301x+3T2017;0;unverified;;2017-11-18 17:57:32;2017-11-18 19:11:22;
```

Figura 26 de Anexo A: Fichero Certificate.csv del curso Android

B Estructura de las tablas auxiliares de la BBDD

En las figuras mostradas a continuación mostraremos la estructura de las tablas auxiliares. En ellas se podrá ver los contenidos, salvo en la tabla relacionada con los países, debido a su extensión y para la imagen mostrada, se decidió recortar. Se debe mencionar que dicha tabla se generó gracias al código libre publicado en <https://github.com/raramuridesign/mysql-country-list/blob/master/mysql-country-list.sql> para obtener así una lista con todos los países y sus respectivas abreviaturas.

Data Output	Explain	Messages	Query History
id	tipo		
integer	[PK] character varying (255)		
1	2 f		
2	1 m		

Figura 27 de Anexo B: Tabla auxiliar Género

Data Output	Explain	Messages	Query History
id	nivel		
integer	[PK] character varying (255)		
1	1 a		
2	10 ap_oth		
3	2 b		
4	3 el		
5	4 hs		
6	5 jhs		
7	6 m		
8	7 none		
9	8 other		
10	9 p		

Figura 28 de Anexo B: Tabla auxiliar Nivel Estudios

Data Output	Explain	Messages	Query History
id	country_code	country_name	
integer	character varying (2)	character varying (100)	
1	AF	Afghanistan	
2	AL	Albania	
3	DZ	Algeria	
4	DS	American Samoa	
5	AD	Andorra	
6	AO	Angola	
7	AI	Anguilla	
8	AQ	Antarctica	
9	AG	Antigua and Barbuda	
10	AR	Argentina	
11	AM	Armenia	
12	AW	Aruba	
13	AU	Australia	
14	AT	Austria	
15	AZ	Azerbaijan	

Figura 29 de Anexo B: Tabla auxiliar de Países.

C Ficheros del componente web

Tras el primer prototipo y la posterior revisión con el cliente de la aplicación, se realizaron como se comenta en el apartado 4.3 una serie de mejoras o se finalizaron ciertas funcionalidades. En la Figura 30 se muestra el directorio dentro de XAMPP donde se aloja la aplicación web con todos los ficheros necesarios para nuestro componente web.

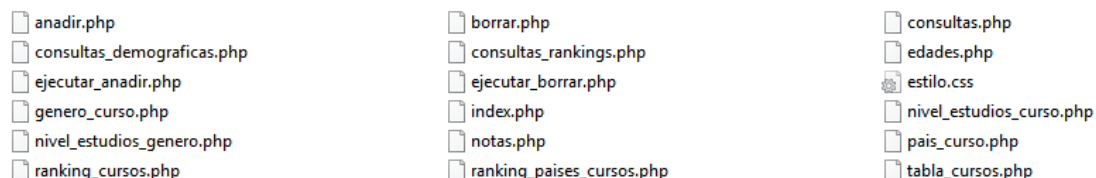


Figura 30 de Anexo C: Directorio de la aplicación uamx del servidor XAMPP.

A continuación, agruparemos los ficheros por funcionalidad dentro de la aplicación y se explicará su utilidad dentro de la misma:

- Gestión de cursos: los ficheros `anadir.php` y `borrar.php` son los que se encargan de añadir nuevos cursos y borrarlos. En `añadir.php` tras haber seleccionado los archivos Certificate y Profile del curso, se usa `ejecutar_anadir.php` para llamar al script desarrollado para el proyecto. Tras haber confirmado en `borrar.php` que quieres borrar los cursos, se ejecuta `ejecutar_borrar.php`, para borrar la base de datos y crearla vacía.
- Consultas: esta agrupación se encarga de realizar las consultas sql sobre la base de datos y mostrar los resultados al cliente en el formato indicado por el cliente en los requisitos funcionales. Haremos aquí también una subdivisión de los ficheros para mostrar con más claridad que hace cada uno:
 - `Consultas.php`: sirve de interfaz para que el usuario pueda elegir que tipo de consulta quiere realizar: si obtener simplemente la tabla con todos los usuarios por curso con toda la información (`tabla_cursos.php`), si realizar consultas de carácter demográfico sobre diferentes aspectos de los alumnos (`consultas_demograficas.php`) u obtener resultados de consultas en formato ranking (`consultas_rankings.php`).
 - `Consultas_demograficas.php`: en este fichero se obtienen por pantalla diferentes inputs de texto o desplegables para realizar diferentes tipos de búsqueda (volver a la Figura 16). Desde esta página, dependiendo que la consulta que quieras hacer se te redirigirá al fichero correspondiente que realizará la consulta y mostrará por pantalla el resultado (`genero_curso.php`, `nivel_estudios_genero.php`, `notas.php`, `edades.php`, `nivel_estudios_curso.php`, `pais_curso.php`)
 - `Consultas_rankings.php`: desde aquí se podrá redirigir a aquellos rankings en los que esté interesado el usuario de la aplicación (`ranking_cursos.php` y `ranking_paises_cursos.php`)
- Página de inicio: al introducir en el navegador la dirección de la aplicación, se llamará siempre a `index.php`. Dicha página muestra una tabla con los cursos que han sido cargados (o un mensaje indicando que todavía no hay ningún curso, en caso de no haber añadido ninguno aún) y se muestran los propósitos y utilidad de la aplicación web.
- Diseño: finalmente, para separar la forma del fondo, se creo el archivo `estilo.css` para dotar de css a la aplicación.

